Reframing AI Discourse

Johnson, Deborah; Verdicchio, Mario

Published in:
Minds and Machines

DOI:
10.1007/s11023-017-9417-6

Published: 01/12/2017

Document Version
Peer reviewed version

Citation for published version (APA):
Johnson, D., & Verdicchio, M. (2017). Reframing AI Discourse. Minds and Machines, 27(4), 575–590.
https://doi.org/10.1007/s11023-017-9417-6

Reframing AI Discourse

Abstract

A critically important ethical issue facing the AI research community is how AI research and AI products can be responsibly conceptualised and presented to the public. A good deal of fear and concern about uncontrollable AI is now being displayed in public discourse. Public understanding of AI is being shaped in a way that may ultimately impede AI research. The public discourse as well as discourse among AI researchers leads to at least two problems: a confusion about the notion of 'autonomy' that induces people to attribute to machines something comparable to human autonomy, and a 'sociotechnical blindness' that hides the essential role played by humans at every stage of the design and deployment of an AI system. Here our purpose is to develop and use a language with the aim to reframe the discourse in AI and shed light on the real issues in the discipline.

1 Introduction

A critically important ethical issue facing the AI research community has to do with how AI research and AI products are responsibly conceptualized and presented to the public. The issue is most evident in the discourse around so-called 'autonomous' technologies. 'Autonomy' is used by AI researchers as a metaphor to refer to a variety of different types of computational behaviour, but the multiplicity of meanings of the term (both for AI researchers and non-experts) can lead to miscommunication: 'autonomy' suggests to those in the media and the lay public something out of human control, something worthy of concern and even fear. In this paper we want to argue for a reframing of AI discourse that avoids the pitfalls of confusion about autonomy and instead frames AI research as what it is: the design of computational artefacts that are able to achieve a goal without having their course of action fully specified by a human programmer. We don't claim that AI researchers have full responsibility for public misunderstanding but we do claim that AI researchers have some degree of responsibility for the way in which their research is presented to and understood by non-experts (the public).

A good deal of concern has recently been expressed about the future of AI research and its consequences for humanity. A salient example is the open letter, signed not only by several AI researchers but also by a number of academics and scientists from other fields, including entrepreneurs, policy makers and professionals. The main point of this letter ("Research priorities for robust and beneficial artificial intelligence" published by the Future of Life Institute. [FLI, 2015a]) is a recommendation to widen the focus of research to include not only the objective of "making AI more capable", but also of "maximizing the societal benefit of AI." The letter writers acknowledge the possibility of AI endeavors that are not beneficial for society or humanity in general. The specter of harmful AI is also evident in some researchers' and entrepreneurs' view of AI. If some see AI as a way for the human mind to overcome the natural decay of the body and live forever in digital form [Itskov, 2016; Minski, 2013; Kurzweil, 2005], others are more keen on warning us against the extinction of the human race by machines that are both stronger and smarter than their creators [Barrat, 2013; Carr, 2014; Storm, 2015; Gaudin, 2015].

Between the extremes of the promise of eternal life and the threat of total annihilation lie the AI artefacts of today: self-driving cars [Google, 2016], package delivering drones [Amazon, 2016], fully automated hedge funds [Aidyia, 2016], to name a few. Every project, whether fully completed or still in development, is accompanied by an array of questions, including compelling ethical questions: Who is to be held responsible when accidents involving self-driving cars occur? [Hevelke and Nida-Rümelin, 2015] If drones can deliver medicines or drop bombs, are we giving life and death powers over humans to artefacts that lack both morality and mortality? [Heyns, 2013; Berkowitz, 2014] When some traders in a stock market are high-speed computers, what is left to do for much slower humans? [MacKenzie, 2014] Will it become impossible for individuals to make informed personal decisions about how to invest their money in financial markets? [Metz, 2016]

All of these questions arise from the simple idea that AI research and products are designed to delegate traditionally human tasks to machines. Hence, the ethical issues all center around the fundamental question: Given task x, what are the consequences of having a machine perform x? This may be considered the most obvious ethical issue in AI, the one to which many researchers are trying to draw attention.

However, the question cannot be answered adequately without better ways of talking and thinking about AI and what happens inside AI artefacts. For one thing, to understand the question as an ethical question requires that it be specified as follows: Given task x, what are the *social* consequences of having a machine perform x? Answering this more specified question requires a conceptual shift that allows the connection between AI and people/society to come into view. All the human actors involved in an AI endeavour must be treated as part of AI, not only the researchers, but those who make the decision to launch AI, those who set up the institutional arrangements in which AI systems operate, and those who fill roles in those arrangements by monitoring, maintaining, and intervening in AI systems.

Others have called for a similar shift. For example, David Mindell illustrates the tight link between humans and technology with several examples of AI artefacts deployed to explore extreme environments such as deep sea and space [Mindell, 2015]. Mindell analyzes existing technologies. We go a step further by using a sociotechnical frame to examine discourse about futuristic AI, Futuristic AI might never come into existence but it is important because discourse about it influences understanding of AI.

In calling for a change in the nature of AI discourse, we are calling for concepts and language that in particular clarify the multiple notions of autonomy that are at play in referring to AI entities as autonomous. What may look like a mere terminological issue reflects a much more serious semantic gap that affects the discussion of AI on several levels and in multiple contexts. The gap misleads AI researchers themselves as well as those in industry, policy makers, and, ultimately, the people whose lives are affected by AI.

2 A New Frame for AI Discourse

Our proposal might be seen as a new ontology because we propose that AI discourse recognize two distinct entities: computational artefacts and AI systems. Computational artefacts are digital entities and AI systems consist of such artefacts together with human actors and social arrangements. Because AI always performs tasks that serve human purposes and are part of human activities, we claim that AI should be understood as systems of computational components together with human behaviour (human actors), and institutional arrangements and meaning. This expanded ontology, we claim, will allow ethical issues to be more readily seen and addressed.

When it comes to computational artefacts we propose a set of distinctions that are quite familiar to AI researchers. Here our purpose is to develop and use language that has the clarity necessary for avoiding (or at least diminishing) confusion and miscommunication about the autonomy of AI. Our point is to demonstrate and emulate the kind of clarity that will allow lay audiences to understand what is and is not possible with AI.

2.1 Computational Artefacts

A *computational artefact* is an artefact whose operation is based on computation. AI researchers are generally focused on a special type of computational artefact, that is, those that are meant to mimic activities that are typically human, such as reasoning, making decisions, choosing, comparing, etc.

Programs in Computers

The first and simplest type of computational artefact is a *program*. Programs receive digital input and produce digital output. The operations of a program remain in the digital realm. Some authors call software an *abstract* artefact [Irmak, 2012], but such characterization better fits algorithms, which are conceived in the minds of human designers and can exist outside the technological realm (e.g. in the form of a block diagram on paper). On the other hand, programs need to be stored in computers in order to operate, so there is a form of embodiment that distinguishes programs from algorithms. Moreover, computers are typically equipped with peripherals that enable them to exchange digital data with other computers (e.g. through network cables) or with humans (e.g. through a keyboard for input, a monitor for output).

Programs in Computers with Sensors

We can distinguish a second type of computational artefact as having a form of embodiment that allows it to receive input from the external environment, that is the non-digital world. Computational artefacts of this kind have sensors. In a way, even a keyboard could be considered a sensor that translates the mechanical movements of a user’s fingers into digital data. This is only partially true: finger movements are simply a non-digital way for a human to insert digital input (i.e. characters and figures) into a computer, whereas here we focus on more sophisticated devices that actually transform a non-digital phenomenon into digital data. Perhaps the simplest example of this kind of entity – one that is often used – is the thermostat of a heating system. The thermostat is connected to sensors that detect temperature; this analog information is translated into digital form so that it becomes input to the program.

Programs in Computers with Sensors and Actuators

A third type of computational artefact both receives input from the external world and *moves* in the external world. We generally call such entities robots. Robots have mechanical parts that allow them to move and, of course, their programs include instructions aimed at controlling those parts. The types of movements that these artefacts can make depend on the forms of the actuators, i.e., their mechanical parts. For example, some robots have wheels allowing them to move across floors, other robots have arms allowing them to reach out and grab, others might have actuators that are weapons. The most successful example of a robot, at least in terms of sales [Morton, 2014], is the Roomba, a robot that cleans floors.

2.2 Autonomy of Computational Artefacts

Humans build artefacts and endow them with the proper hardware and software with specific goals in mind. By delegating the execution of the operations needed to reach those goals to the artefact, humans are freed of that burden.

This is the basic idea behind automation. It characterizes all sorts of artefacts including computational ones. Humans are happy to delegate tasks to computational artefacts since they are able to execute operations at super-human speed without errors. If all computational artefacts are automatic, what makes some of them ‘autonomous’? What does it mean for an artefact to be ‘autonomous’?

Let’s start with an example from the first category, a program in a computer, and imagine a software agent for trading that is supposed to connect to a server and buy shares from the best company available. The criteria to compare companies and establish the best one are fully coded into the agent, but there may be the possibility that two or more companies have exactly the same best parameters. The designers could write the software in a way that, in such a situation, it sends a message to the human user on behalf of whom the agent is operating. The agent will then buy shares from the company indicated by the user. A different way to implement the agent is to write its code in a way that, faced with the above-mentioned decision, it will perform a sequence of operations that makes the selection of the company possible without human intervention. The designers have many choices on how to implement the selection process: among the eligible companies, the agent could pick the first one in alphabetical order, or the oldest, or the newest, and so on. The agent might even pick the company, metaphorically speaking, by means of a “coin toss”, that is, based on the value of a randomly generated number.

At first glance, the trading agent that does not require human intervention for the purchase of the shares appears to be more ‘autonomous’ than the other. This is true but not the whole picture. If autonomy in programs means simply no human intervention, then software written to print the first one hundred prime numbers on a screen would have to be considered autonomous since it does not require any human intervention during its run.

So, we need a more precise account of autonomy in programs. In the prime numbers printer, the execution is entirely established already *at compile time*, i.e. when the code is written by a human designer, step by step from beginning to end. By contrast, in the case of our trading agents, the course of action is established *at run time* and depends on the data coming from the server the agent connects to. In the case of the agent that comes back to its user to ask for a decision, at least one of those run time conditions is an action/intervention by the user, while the other agent will base its decisions solely on the basis of what is written in its code and the data from the server.

It seems, then, that autonomy is a characteristic of artefacts in which the course of action is established at run time, without human intervention and on the basis of the conditions in the environment in which the artefact operates. Artefacts that require human decision at run time – as with the trading agent requiring a user to choose between two best companies – are less autonomous than the ones that require no human input at run time.

From this perspective, endowing an artefact with sensors seems to increase its autonomy, because the sensors decrease the need for human intervention. For example, compare an artefact that triggers the watering of a garden at regular intervals with another that acts on the basis of the level of humidity of the terrain as measured by means of sensors. The owners of the garden need to intervene in the operation of the time-based artefact in at least two possible ways: they have to switch it off if there has been abundant rain, and they have to manually activate it for extra water during particularly hot and dry days. On the other hand, with an artefact endowed with sensors, the owners are freed of the burden of intervention/control: the artefact will see to it that the humidity of the terrain is always at the optimal level, independent of the owner’s monitoring of the weather.

In the same way, the addition of actuators further increases autonomy. Imagine the garden-watering artefact in the form of a robot with arms and wheels: it could be programmed to move around all the gardens in the area, check the levels of humidity in each terrain, and obtain and carry water or turn on nearby spigots where needed. Even more autonomy could be achieved by means of additional peripherals, be they sensors or actuators.

Needless to say, the additional peripherals would require additional code to enable the artefact to use the peripherals. The additional code would elaborate the input from the additional sensors and control the movements made possible by the additional actuators.

Autonomy is, then, a function of how a computational entity operates at run time when it draws on input from the environment. The entity's autonomy has to do both with its responsiveness to its environment and independence from human intervention at run time and may also be a function of increased capacity for movement. The less intervention needed by humans in its operation and the wider its scope of action, the more autonomous the artefact.

2.3 Unpredictability of Artefacts

When autonomy is understood in this way, it becomes clear that people will likely pay less attention to the artefact's operation. This makes the artefact more unpredictable. Imagine again the garden-watering robot. If its owners realize that it is not in their garden, they may (correctly) think that the robot must be watering some other garden in the area. However, they would not be able to predict where exactly it is. If human users do not observe the artefact at work, and this happens often when the artefact is supposed to be autonomous, they will not know what kind of input the artefact received, hence it will be difficult, if not impossible, to predict how the artefact will operate to achieve its goal.

There are many ways in which artefacts can be unpredictable. Consider some examples. The random number generator used by the trading agent mentioned before is a piece of software that applies a complex mathematical function to data provided by the computer's clock. The output of such software is a sequence of numbers that seem not to have been determined by any mathematical function, and thus appear to be randomly chosen by the software. Obviously, such a function exists because a computer only operates through functions and mathematical operations, but if a user does not know what the function, the output will indeed look random. Even the programmers who designed the system are not able to foresee the numbers in the output because the function is parametric and its results depend not only on the function itself but also on numerical values from the clock, such as the milliseconds of the time at which the software was launched. If this piece of information is missing, not even the designer of a random number generator can predict its output.

This is a very important point that deserves attention: human users of computational artefacts, including their designers, need a certain amount of information to be able to predict the course of action of the artefacts. Some artefacts are such that one only needs to know its code to predict the outcome (e.g. the prime number printer), whereas other artefacts require observation throughout their run in order to make predictions on how they will operate. For instance, in the case of our software agent purchasing the best companies, we would be unable to predict its behaviour unless we could know the situations of all the companies on the stock market. In principle, however, if on our own we were able to find out which company is the best (using the same criteria as the agent), we could anticipate (predict) that our trading agent will buy shares of that company, provided that the software is not faulty.

In order to predict which of two top companies our trading agent will choose in the event of a tie, we will have to know the specific criteria that are coded into the trading agent. If the criteria are based on the names of the companies or the years of their founding, we need to acquire this information to know what the agent will do. Now suppose that the agent is designed to use a random number generator: it runs the generator to pick a number between 0 and 9; if the output is between 0 and 4 the agent will buy shares from the first company, whereas if the output is between 5 and 9 it will buy from the second company. In this case, we will need a different kind of information to be able to predict the outcome: as said before, we need to know what the mathematical function used in the generator looks like and the exact time at which the generator was launched. Since this last piece of information is extremely difficult to acquire, it is likely that we will fail at our task, and that the agent's decision will have the appearance of a random act.

Indeed, random number generators exist that, just like a thermostat, rely on events that happen in the external environment: a computational artefact can be endowed with a light sensor that contains a "beam splitter", that is, a half-mirror that splits light in two orthogonal rays. The device includes two photon sensors that can detect where each photon from the split ray goes: one way or the other, according to which a 0 or a 1 will be generated by the device. The trading agent may be implemented so to buy shares from the first company in case of a 0, and from the second company in case of a 1. Unpredictability is increased here in the sense that no human can predict where a photon will go (i.e. a quantum mechanical phenomenon) and, thus, which figure will be generated. This is why devices that are based on physical phenomena are called 'true' random generators as opposed to computational 'pseudo'-random generators [Jennewein et al., 1999]. This is the fundamental mechanism that enables software engineers to create programs that operate stochastically: they have the possibility to make the completion of an instruction depend on the result of a pseudo-random or a truly random event.

This kind of operation is often used to randomly explore different possibilities, in search of an optimal solution. Google, for instance, has set up some experiments to train robotic arms in the task of opening a door. The computers controlling the arms have been provided with code with commands that should roughly guide the hardware with the right moves. Every time these commands are executed, a random small numerical value is added to the parameters that determine the positions of the parts of the robotic arms, resulting in new, slightly different movements at each round. The movements with the best outcome are then registered in the system for future use [Levine et al., 2016].

Lack of information on behalf of the human users make computational artefacts unpredictable, but the unpredictability stems from several different kinds of ignorance: ignorance of the functions used or of the time of their activation (as in the pseudo-random number generators), impossibility of predicting quantum mechanical phenomena (as in the true random number generators), or simply ignorance of the circumstances in which the artefact operates (as when we cannot predict which shares the trading agent will buy if we do not know the market, or the whereabouts of the garden-watering robot if we have not been observing it).

Since autonomous artefacts need little to no human intervention at run time, indeed, since they are often conceived to free humans from the burden of several tasks, it should not be surprising that users do not have a full knowledge of the environment in which the artefacts operate. Hence, the autonomy of artefacts is linked to their unpredictability. Computational artefacts are unpredictable because humans don't and can't know the input on which the operation of the artefact depends.

2.4 Limits to Unpredictability

The unpredictability of computational artefacts is important for our purposes here because, rightly or wrongly, it plays into public fear and concern about 'autonomous' machines. However, it is important to note that the unpredictability of the operations of an artefact, even when intrinsic because based on quantum mechanical phenomena, is limited by at least two factors. Firstly, the designer had to specify the kind of analog input that could be received by the artefact: the choice of endowing the artefact with a temperature sensor or with a light sensor determines what kind of environmental factors will influence the operations of the device. Secondly, whatever the randomness in the input that affects the operation of the artefact, the range of its course of action is bounded by its actuators which in turn are bounded by the set of operations specified by human designers, i.e. the operations that control the capabilities of the artefact.

A Roomba, for example, is 'autonomous' in the sense that its course of action (e.g. in terms of movements of its wheels) at any given moment depends on the input it receives about the environment and because this input is used, in accordance with the robot's software, to compute subsequent movements. Although the movement of the Roomba is unpredictable (because so is the input from the environment and an average Roomba user does not know its internal computations), one can, nevertheless, predict (and be confident) that the Roomba will not behave in certain ways. For example, we know the Roomba will not climb up the walls or fly because we can see that it doesn't have the mechanical parts necessary for such behaviour. Moreover, if we had the possibility to examine its software and saw that nowhere in its code was an operation to compute the square root of 2, then we would be able to predict that the Roomba will never perform such an operation.

Unpredictability is often thought to occur or increase when software is programmed to learn. *Learning* can play a significant role in seeming to expand the autonomy of computational artefacts. If the artefact is able to acquire new patterns of behaviour by means of proper training, then the system's autonomy may increase over time. Imagine a futuristic Roomba whose hardware includes a camera able to capture an image of every object the robot is about to suck up, and a sensor that detects when an object is too big and will likely clog the robot's mouth. With the proper software, including instructions to compare the current input of the camera with stored images of previously encountered objects, this Roomba might learn to avoid certain objects just like it already avoids furniture. Moreover, a Roomba might learn by receiving negative feedback from its owner (e.g. because it has sucked up a piece of Lego that was supposed to stay on the floor). The negative feedback takes the form of new inputs for the operation of the learning software.

Nevertheless, even when robots learn in this way, their autonomy is a matter of programmed instructions – instructions that may make the behaviour of the robot difficult for some to predict, but not difficult to predict in the sense that the behaviour will be within the boundaries specified in the program as well as the boundaries of the hardware. Even in an extreme case of unpredictable results like Microsoft's Twitter-bot (a learning software that was taken offline because it had learned racial slurs from Twitter users and started tweeting them around), the unpredictability was limited to the content of the tweets (e.g. the software did not learn new actions like accessing internet banking services). Its learning racial slurs might have been avoided if designers had tested the program for this quality or observed it more carefully when it was first operating.

So, autonomous computational artefacts have a certain kind of unpredictability that is related to their autonomy. However, because their unpredictability derives from the limitations of human users and observers, it is important to remember that autonomous computational artefacts are still bounded by their programming – even when they learn – and their embodiment.

2.5 AI Systems

So far our analysis of autonomy has focused on computational artefacts. Indeed, most of the literature on autonomous systems focuses on this component of AI. However, AI that perform tasks on behalf of humans consist of much more in addition to computation by artefacts. We propose that the ontology of AI discourse be expanded to include *AI systems*. An AI system consists of a computational artefact together with the human behaviour and people who make the artefact a useful and meaningful entity. Drawing on a concept and a term from the field of Science and Technology Studies (STS), AI systems should be thought of as *sociotechnical ensembles* [Bijker, 1993; Bijker, 1997] or sociotechnical systems. Sociotechnical ensembles are combinations of artefacts, human behaviour, social arrangements and meaning. For any computational artefact to be used for a real-world purpose, it has to be embedded into some context in which there are human beings that work with the artefact to accomplish tasks. Human actors may be required to launch (turn on) the computer in which the computational artefact resides, monitor the artefact's operation, give it input, use the output, and so on. Moreover, the artefact will have meaning to the humans involved. Imagine here an extremely well designed AI program for a new form of monetary exchange, e.g., bitcoin, airline miles. Unless the program is connected to other computers, it has no real-world functionality. Moreover, for it to become a new monetary system, networks of people have to recognize computer configurations in the system as having value, and they have to accept these configurations as a form of money [Johnson and Miller, 2008].

Human actors might be understood to be part of the external environment of AI in that they give input to the computational artefact. However, what humans do is more than that. For example, a drone that has been programmed to select targets and fire under certain conditions will be part of a military operation. In the military operation, humans will decide when to launch the drone and what initial input to give to the drone; humans will monitor the drone and decide if and when to change its instructions or when to have it return to the home base. Even if decisions to change instructions or return to home base are programmed in, a person has to decide whether or when to launch a drone and in what conditions or context. Moreover, a strike by a drone counts as an act of war because of the meaning associated with such behaviour by institutional actors (e.g. the governments of the nations at war). Indeed, recent conflicts in which drones were used have taught us that drones have different meanings in different cultural contexts [Ahmed, 2014].

The design of AI systems like the design of other sociotechnical systems involves decisions about how to delegate sub-tasks among humans and non-humans [Latour, 1992; Callon, 1999]. Taking a very simple example, when it comes to heating a building, the furnace is assigned certain tasks and the thermostat others. These components work together with humans who have been delegated the task of deciding where the controls will go and the task of setting the temperature on the thermostat, not to mention those who manufacture and install the device. Even in an office building, where individuals cannot control the temperature in their own offices, a maintenance person may control the temperature. Of course, this might be done with a program, but even here a person would have to set the parameters of the program.

Unquestionably, more and more tasks are being delegated to computational artefacts and that is why it is so important to remember that humans are always part of the system.

3 Confusion about Autonomy

Given what has been said about computational artefacts, the fear and concern being expressed in the public discourse about AI do not seem justified, or more accurately, the fear and concern seem misdirected since the behaviour of computational artefacts is in the control of the humans that design them. The range of possible outputs in a computational artefact, even those with sensors and actuators and embedded in social arrangements, are specified by the parameters in the instructions of the program and are limited both by the programming and the limitations of the hardware.

So, why such public fear and concern? Those who don't understand how computers work have a very different notion of autonomy, one that is associated with human beings (in normal conditions). Here autonomy refers to the characteristic of human beings of having the capacity to make decisions, to choose, and to act. 'Autonomy' is here tied to ideas about human freedom. This notion of autonomy has traditionally been used to distinguish humans from other types of animals. Importantly, this form of autonomy is what makes human beings moral beings. Only beings with autonomy can be expected to conform their behaviour to rules and laws. Indeed, when it comes to morality a distinction is made between entities that behave according to the laws of nature (e.g., the leaves of a tree turning towards the sun) and entities that behave according to the conception of law (e.g., a person choosing to keep a promise or tell the truth or not) [Kant, 1785]. Admittedly, this form of autonomy is somewhat mysterious and is intertwined with notions of what it means to be human. Nevertheless, it is this notion of autonomy that seems to come into play in the fear and concern about autonomous machines or robots. When non-experts hear that machines have autonomy, they attribute to machines something comparable to the autonomy that humans have, something close to the freedom to behave as one chooses.

When the public, the media, and anyone who is not familiar with the workings of computers is told that machines have autonomy, it conjures up ideas about an entity that has freewill and interests of its own – interests that come into play in decision making about how to behave. They infer that programming will be insufficient to control such entities, that is, to ensure that they will behave only in specified ways. Such entities will, they fear, behave in unpredictable ways, i.e., ways that serve their own interests.

Although human autonomy may in certain contexts be a useful metaphor for the autonomy of computational artefacts, some scholars get caught up in the metaphor and seem to forget the difference between the thing and its metaphorical parallel. An example of this can be seen in Omohundro's chapter in *Risks of Artificial Intelligence* (2016). In describing the possible harmful behaviours of an advanced AI, Omohundro adopts the approach of presenting scenarios in which an artefact behaves like a (possibly sociopathic) person who harms others in the blind pursuit of its own objectives. He describes, for example, a chess-playing robot and a human trying to unplug it: "*Because nothing in the simple chess utility function gives a negative weight to murder, the seemingly harmless chess robot will become a killer out of the drive for self-protection.* [Omohundro, 2016, p.15]" The drive for self-protection, a natural characteristic of humans and many other biological entities, is presented by the author as a property of advanced AI artefacts. The drive is then supposed to lead to resource acquisition behaviour: "*The chess robot* (...) *would benefit from additional money for buying chess books* (...) *It will therefore develop subgoals to acquire more computational power and* *money. The seemingly harmless chess goal therefore motivates harmful activities such as breaking into computers and robbing banks.* [ibid. p.16]". Omohundro also attributes other human properties to machines, for example, the drive for efficiency: "[Autonomous systems] *will aim at making every joule of energy, every atom, every bit of storage, and every moment of existence count for the creation of expected utility* [ibid. p.17]". Something similar is done with the drive for self-improvement: "…*autonomous systems will be motivated to completely redesign themselves to take better advantage of their resources in the service of their expected utility* [ibid. p.17]."

Omohundro attributes to the chess playing program a set of characteristics that are associated with the behaviour of humans. He uses language and concepts used in talking about humans. He leaves entirely out of the picture that, at the current state of technological development, something like self-protection in the robot would be produced computationally through instructions given to it by humans. Attributing to robots the quality of self-protection is a metaphor. It is like saying "let's treat this chunk of computational behaviour as if it were something like self-protection in humans." This way of discussing AI has the purpose of depicting a possible future scenario, but it is misleading and dangerous insofar as it distorts what is currently possible in AI thereby suggesting to non-experts that some dangerous form of computation is in the making.

Omohundro is, of course, speculating and extrapolating from the current state of computation to the future but does not bother to explain that the kind of robotic behaviour he envisions would require computational forms of a radically different kind from current computation. Although the public comes to believe such scenarios are possible, the possibility of such new computational forms is neither probable nor improbable, but simply unknown.

The absence of any real understanding of how imagined, futuristic robots will work gives futuristic thinkers a free hand to present misleading and sometimes contradictory scenarios. Here is an example of a futuristic superintelligent machine designed with the directive to "*make all people happy* [Yampolskiy, 2016, p.131]" proposed by Yampolskiy. Among the many alternative ways that such a machine could 'autonomously' calculate to reach its goal, the author lists killing all people, performing lobotomies, affixing permanent smiles by means of forced plastic surgeries, a daily dose of ecstasy. According to Yampolskiy, the machine has an infinite number of approaches to choose from, and the chosen one "*may be anything but desirable for humanity*. [ibid. p.132]" He seems to forget to mention that, in existing machines, *choosing* means computational processes that are programmed by giving the machine instructions, and that for that choice to become something comparable to the freedom of action that human beings have, a radical technological breakthrough (something like Kurzweil's *singularity* [Kurzweil, 2005]) must occur. Whether or not such technological advancement will be possible in the future, its hypothetical results will have to be substantially different from AI systems of today.

Futuristic thinking has an important role to play in the development of new technologies – in stimulating thinking about what is possible and what new technologies might mean. We might take these AI scenarios to be cautionary tales about how not to design AI. However, many of the descriptions of this kind are irresponsible insofar as they hide how computational artefacts actually work and how the workings of the hypothetical artefacts of the future are as yet unknown and, in fact, impossible with the kind of computing available today and for the foreseeable future.

4 Sociotechnical Blindness

Absence of discussion of the role played by programmers and other human actors in creating AI is another problem in current AI discourse that leads to misunderstanding and fear. What we call *sociotechnical blindness*, i.e. blindness to all of the human actors involved and all of the decisions necessary to make AI systems, allows AI researchers to believe that AI systems got to be the way they are without human intervention. As with confusion about autonomy, this blindness facilitates futuristic thinking that is misleading. It entirely leaves out of the picture the fact that to get from current AI to futuristic AI, a variety of human actors will have to make a myriad of decisions. Human actors will have to decide what sort of AI research to invest in, what kind of parameters to put into the instructions of programs, what kind of hardware to develop and connect up to computers. Human actors will have to decide what contexts to embed the artefacts in and what social arrangements to set up to launch, monitor, and maintain the artefacts. Moreover, in order to get to a future in which computational artefacts exhibit behaviour that might be called 'kind', 'malicious' or 'self-preserving', human actors will have to agree (implicitly if not explicitly) to use language in that way. They will have to accept the use of these terms when applied to computational entities.

4.1 (Un)Predictability

Neglecting the human actors in the development of a computational artefact makes the artefact seem more unpredictable than it actually is. Let us consider again Omohundro's chess-playing killer robot, and let us compare it to the Roomba, which is a current system that is autonomous according to our definition. Even if the chess-playing killer robot has much more advanced and complicated programming, if its operations are regulated by the same basic principles as the Roomba's, our analysis of the limitations to the unpredictability of the Roomba also apply to the futuristic robot.

Imagine questions about the possibility of the futuristic chess-playing artefact unpredictably killing a human. Is such an event possible? Omohundro himself asks: why would a chess-playing robot kill the human who is trying to shut it down? His answer is that such an act might turn out to be in accordance with its goal of maximizing its utility function: the robot will take any possible action to be able to play chess. This answer is wholly misleading because a chess-playing robot would be directed at playing chess. To imagine that the chess-playing robot could do more than make moves on a chessboard requires that we imagine the robot to have been built with sensors and actuators that detect and operate on embodied human behaviour. Aside from the fact that this would likely be well beyond what would be required to play chess, if the chess playing robot did have the sensors and actuators necessary to kill, they would have had to have been put there, that is, put into the software (programming) and hardware of the robot.

With regard to the software we can ask: where does the chess-playing robot's goal come from? Either it was provided by a human programmer or, in a futuristic scenario, by another machine, which, in turn, was designed either by a human or another machine, and so on. The origin of the drive guiding the operation of the artefact can always be traced back to the choice of a human designer. Is the designer aware of the fact that the robot is going to play chess no matter what, even at the cost of a human life? If so, then it would seem the designer would be irresponsible in building particular sensors and actuators into it and then unleashing such a robot on innocent chess players.

Yampolskiy's futuristic example of a robot killing because it is directed at happiness suffers from the same sociotechnical blindness. He warns that humans might set an artefact's goal and the artefact might try to reach it in harmful ways. However, is it possible to have pre-established goals attained in unpredictable ways? In other words, could a chess-playing robot become a killer robot? As already suggested, what kind of actuators would it have to be endowed with? Would it have a gun attached to its body? Would it have an arm with which it could grab and use a knife? If so, would its software include instructions to control these mechanical parts? For a robot to pull a trigger, the relevant instruction must be in the program controlling its behaviour. Such instruction must have been written in the robot's memory, either directly by a programmer, or indirectly by means of machine learning. Even if these 'superintelligent' machines of the future can learn at unprecedented speeds, in order for them to act, a command must be present in their software and the command must be connected to embodied actuators. To think otherwise is to fall into an even greater fallacy than the autonomy confusion, because it involves imagining that such machines not only can act the way humans do, they can even conjure up acts out of nothing. If this is what 'superintelligence' is about, then it is nothing short of magic!

By keeping in sight the human actors who make AI systems what they are, the connection between the seeming unpredictability of artefacts and the issue of responsibility becomes much clearer, and fallacies like the so-called "responsibility gap" can be effectively countered.

4.2 Responsibility

The responsibility gap is a concept introduced by Matthias to describe a situation in which no person can be held responsible for the consequences of the behaviour of an artefact [Matthias, 2004]. Instead of depicting far-fetched futuristic scenarios, Matthias focuses on the possible development of existing artefacts, like the AIBO dog-shaped robot. He writes: "*With a little experimentation* [the AIBO] *will be able to find out that its battery life can be prolonged by galloping…the robot, while running around the apartment, collides with a small child and injures him.* [ibid. p.177]" According to the author, this is an "*unforeseeable*" development for which nobody can be justly said to be responsible. However, from the perspective of responsibility, the same designers who programmed the robot to have the goal to save battery power and endowed it with the capability to gallop should have added also programmed in the goal to avoid obstacles. Matthias has hidden from view the human actions (or inactions) that would be necessary to produce the dangerous AIBO.

The same ascription of responsibility can be applied to Omohundro and Yampolskiy's killing machines. However, the point is not to play the blame game, especially because such scary artefacts do not exist (yet, according to these authors). Rather, the point is that putting into the world a robot that has the capability of harming humans is a human act, and the human actors who release such computational artefacts will be responsible for the consequences, not the computational artefact itself. This shifts the focus from futuristic computational artefacts to those who design and build them and embed them in social contexts.

5 Conclusions

In this paper we began with the idea that there is an ethical issue with regard to how AI researchers conceptualize, talk about, and present AI. We have argued that discourse about AI leads to misunderstanding and ultimately fear of AI because of two problems in the way AI is discussed and presented. The first problem is confusion about autonomy and the second is blindness to the human actors and human behaviour that are part of AI systems. We have tried to show that these problems can be tackled by distinguishing AI computational artefacts and AI sociotechnical systems, which include computational artefacts. When this shift in thinking is made the nature of autonomy in AI systems can be clarified and the human actors who are an indispensable part of AI systems can be kept in sight. Our claim is that AI research and researchers will be better served and will provide better public understanding of AI by framing the discourse in this way.

From this perspective, a document like the open letter issued against the indiscriminate use of autonomous weapons [FLI, 2015b] makes much more sense than expressions of fear about the so-called uncontrollability of future AI. The letter warns that this new kind of artefact might be extremely harmful if it ends up in the wrong hands. This is another way of saying that we should be concerned about the human actors (and their autonomy) who are part of AI systems. Who is deciding which AI systems to build and put in a context? Who is deciding and how are decisions being made about which tasks to delegate to humans and which to machines? How are the humans that work within AI trading systems, self-driving transportation systems, or drone systems being trained? Indeed, there are many reasons for concern and even fear about autonomous systems, but these reasons have to do with the human actors in AI systems and not merely the computational artefacts in them.

References

[Ahmed, 2014] Akbar Ahmed. *The Thistle and the Drone: How America's War on Terror Became a Global War on Tribal Islam.* Harper Collins Publishers India, Noida, India, 2014.

[Aidyia, 2016] Aidyia. *Aidyia: About us*. Retrieved from www.aidyia.com/company/, 2016.

[Amazon, 2016] Amazon. *Amazon Prime Air*. Retrieved from www.amazon.com/primeair/, 2016.

[Barrat, 2013] James Barrat. *Our Final Invention: Artificial Intelligence and the End of the Human Era.* Thomas Dunne Books, New York City, NY, 2013.

[Berkowitz, 2014] Roger Berkowitz. Drones and the Question of "The Human". *Ethics & International Affairs*, 28(2):159–169, 2014.

[Bijker, 1993] Wiebe E. Bijker. Do not despair: there is life after constructivism. *Science, Technology & Human Values*, 18(1):113–138, 1993.

[Bijker, 1997] Wiebe E. Bijker. *Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change.* MIT Press, Cambridge, MA, 1997.

[Callon, 1999] Michel Callon. Actor-network theory—the market test. *The Sociological Review* 47(S1):181–195, 1999.

[Carr, 2014] Nicholas Carr. *The Glass Cage: Automation and Us.* W. W. Norton & Company, New York City, NY, 2014.

[FLI, 2015a] Future of Life Institute. *Research priorities for robust and beneficial artificial intelligence*. Retrieved from http://futureoflife.org/ai-open-letter/, 2015.

[FLI, 2015b] Future of Life Institute. *Autonomous Weapons: An Open Letter from AI & Robotics Researchers*. Retrieved from http://futureoflife.org/open-letter-autonomous-weapons/, 2015.

[Gaudin, 2015] Sharon Gaudin. Stephen Hawking fears robots could take over in 100 years. *ComputerWorld*, 14 May 2015.

[Google, 2016] Google. *Google Self-driving Car Project*. Retrieved from www.google.com/selfdrivingcar/, 2016.

[Hevelke and Nida-Rümelin, 2015] Alexander Hevelke and Julian Nida-Rümelin. Responsibility for Crashes of Autonomous Vehicles: An Ethical Analysis. *Science and Engineering Ethics*, 21(3):619–630, June 2015.

[Heyns, 2013] Christof Heyns. *Report of the Special Rapporteur on Extrajudicial, Summary or Arbitrary Executions, Christof Heyns*. United Nations Human Rights Council, session 23, 9 April 2013.

[Irmak, 2012] Nurbay Irmak. Software is an Abstract Artifact. *Grazer Philosophische Studien*, 86:55–72, 2012.

[Itskov, 2016] Dmitry Itskov. *2045 Strategic Social Initiative*. Retrieved from http://2045.com, 2016.

[Jennewein et al., 1999] Thomas Jennewein, Ulrich Achleitner, Gregor Weihs, Harald Weinfurter and Anton Zeilinger. A Fast and Compact Quantum Random Number Generator. Retrieved from arxiv.org/abs/quant-ph/9912118, 1999.

[Johnson and Miller, 2008] Deborah G. Johnson and Keith W. Miller. Un-making Artificial Moral Agents. *Ethics and Information Technology*, 10(2-3):123–133, 2008.

[Kant, 1785] Immanuel Kant. *Groundwork of the Metaphysics of Morals*, 1785.

[Kurzweil, 2005] Ray Kurzweil. *The Singularity is Near: When humans transcend biology*. Penguin Books, London, UK, 2005.

[Latour, 1992] Bruno Latour. Where are the Missing Masses? The Sociology of a Few Mundane Artifacts. In Bijker & Law (eds.) *Shaping Technology/Building Society: Studies in Sociotechnical Change*. MIT Press, Cambridge, MA, 1992.

[Levine et al., 2016] Levine, Sergey, Peter Pastor, Alex Krizhevsky and Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. Google Preliminary Report available at arxiv.org/pdf/1603.02199v4.pdf, 2016.

[MacKenzie, 2014] Donald MacKenzie. *A Sociology of Algorithms: High-Frequency Trading and the Shaping of Markets.* Retrieved from http://www.sps.ed.ac.uk/__data/assets/pdf_file/0004/156298/Algorithms25.pdf, 2014.

[Matthias, 2004] Andreas Matthias. The Responsibility Gap: Ascribing Responsibility for the Actions of Learning Automata. *Ethics and Information Technology*, 6(3):175–183, 2004.

[Metz, 2016] Cade Metz. The Rise of the Artificially Intelligent Hedge Fund. *Wired*, 25 January 2016.

[Mindell, 2015] David A. Mindell. *Our Robots, Ourselves: Robotics and the Myths of Autonomy*. Viking Press, New York City, NY, 2015.

[Minski, 2013] Marvin Minski. *Dr. Marvin Minsky – Facing the Future*. Retrieved from www.youtube.com/watch?v=w9sujY8Xjro, 2013.

[Morton, 2014] Oliver Morton. Good and ready. *The Economist*, 29 March 2014.

[Omohundro, 2016] Steve Omohundro. Autonomous Technology and the Greater Human Good. In Vincent Müller (ed.) *Risks of Artificial Intelligence*, 9–27. CRC Press, Boca Raton, FL, 2016.

[Storm, 2015] Darlene Storm. Steve Wozniak on AI: Will we be pets or mere ants to be squashed our robot overlords? *ComputerWorld*, 25 March 2015.

[Yampolskiy, 2016] Roman V. Yampolskiy. Utility Function Security in Artificially Intelligent Agents. In Vincent Müller (ed.) *Risks of Artificial Intelligence*, 115–140. CRC Press, Boca Raton, FL, 2016.